

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年11月10日

出 願 番 号

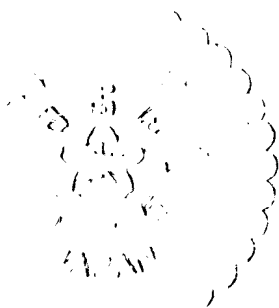
Application Number:

特願2000-342848

出 願 人

Applicant(s):

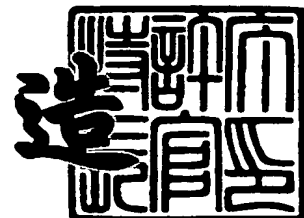
三洋電機株式会社



2001年11月 2日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3095910

【書類名】 特許願

【整理番号】 KIA1000078

【提出日】 平成12年11月10日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 3/00  
G08C 15/06

【発明者】

【住所又は居所】 大阪府守口市京阪本通 2 丁目 5 番 5 号 三洋電機株式会社  
社内

【氏名】 近藤 英雄

【特許出願人】

【識別番号】 000001889

【氏名又は名称】 三洋電機株式会社

【代表者】 桑野 幸徳

【代理人】

【識別番号】 100111383

【弁理士】

【氏名又は名称】 芝野 正雅

【連絡先】 0 3 - 3 8 3 7 - 7 7 5 1 法務・知的財産部 東京事  
務所

【手数料の表示】

【予納台帳番号】 013033

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9904451

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 マイクロコンピュータの制御方法

【特許請求の範囲】

【請求項 1】 ホストとマイクロコンピュータ間のデータ送受信のインターフェースを行う USB インターフェース回路と、書き込み制御プログラムが格納された第 1 のプログラム領域及びプログラムデータが書き込まれる第 2 のプログラム領域とを有し、電氣的に書き換え及び読み出し可能な不揮発性メモリと、前記不揮発性メモリから読み出されるプログラムを実行する CPU と、を備えたマイクロコンピュータの制御方法であって、

マイクロコンピュータのリセットに応じて前記書き込み制御プログラムの実行を開始する第 1 の段階と、前記 USB の初期化を実行する第 2 の段階と、前記ホストからプログラムデータを前記不揮発性メモリの第 2 のプログラム領域に書き込む第 3 の段階と、前記第 2 のプログラム領域に書き込まれたプログラムを実行する第 4 の段階と、を有することを特徴とするマイクロコンピュータの制御方法。

【請求項 2】 前記マイクロコンピュータのリセットはパワーオンリセットであることを特徴とする請求項 1 に記載のマイクロコンピュータの制御方法。

【請求項 3】 前記第 4 の段階の後に、前記ホストからプログラムデータを再書き込みするか否を判定する段階を有し、再書き込みする場合には前記第 1 の段階から第 4 の段階を再度繰り返すことにより、前記ホストからプログラムデータを前記不揮発性メモリの第 2 のプログラム領域に再書き込みし、当該再書き込み後のプログラムを実行することを特徴とする請求項 1 に記載のマイクロコンピュータの制御方法。

【請求項 4】 前記第 3 の段階の後に、前記第 2 のプログラム領域内の特定の領域に照合用コードデータを書き込む段階を有すると共に、前記第 1 の段階の後に、前記特定の領域のデータと照合用コードデータとを比較する段階を有し、両者が一致しない場合には前記第 3 の段階に移行し、両者が一致した場合には前記ホストからプログラムデータを前記不揮発性メモリの第 2 のプログラム領域に再度書き込むことなく前記第 2 のプログラム領域に書き込まれたプログラムを実行

することを特徴とする請求項 1 に記載のマイクロコンピュータの制御方法。

【請求項 5】前記ホストはパーソナルコンピュータであることを特徴とする請求項 1、2、3 及び 4 に記載のマイクロコンピュータの制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、USB インターフェース回路及びプログラム格納用メモリとしてEEPROMを内蔵したマイクロコンピュータの制御方法に関するものであり、特にホスト（例えば、パーソナルコンピュータ）からEEPROMへのプログラムデータの書き込み及び実行に係る制御方法に関する。

【0002】

【従来の技術】

近年、パーソナルコンピュータ等において、周辺デバイスの拡張性の自由度を高めるために、USB (Universal Serial Bus) のサポートが始められている。USBはユーザの利便性を考慮して考案されたシリアルインターフェース規格であって、キーボード、マウス、カメラ、プリンタ、スキャナー、スピーカ等の様々な周辺デバイスとパーソナルコンピュータ等との通信に共通に使用できる。

【0003】

図6はUSBを利用したパーソナルコンピュータと周辺デバイスとの接続構成例を示す図である。上位のパーソナルコンピュータ100とハブ101との間はUSBケーブルで接続され、さらにハブ101の下位には周辺デバイス102～105が接続され得る。そして、パーソナルコンピュータ100によって周辺デバイス102～105の管理が行われる仕組みになっている。このように、USBは多重スター型のネットワーク構造の双方向通信可能なシリアルバスといえる。

【0004】

ここで、USBケーブルには4本の信号線が含まれる。その内訳は電源用2本と、データ信号用2本である。データ信号は基本的には差動信号 ( $D^+$ ,  $D^-$ ) として扱われる。また、USBを利用したデータ転送は、転送単位がフレームとい

う概念で時間分割され、そのフレームを積み重ねていくことにより行う。1つのフレームはS O F (Start Of Frame) パケットにより開始する。そして、ホストのパーソナルコンピュータは予めそのフレームの中にスケジューリングされたデータ転送要求トークン(キーボードやカメラからのデータ入力要求や、音声データの出力要求)を順次送出することにより、複数の周辺デバイスとのデータ転送を並行して行う。

【0 0 0 5】

なお、USBに関する技術文献として、例えば「Interface」(1997年1月号)、特開平11-205412号公報等がある。

【0 0 0 6】

ところで、上述した周辺デバイスには、通常、デバイスの動作を制御するためのマイクロコンピュータが搭載される。ここで、マイクロコンピュータにはプログラム格納用メモリとして、電氣的に書き換え、読み出し及び消去可能な不揮発性メモリであるEEPROMが内蔵されているものとする。このEEPROMには、上記の機能に加えてプログラムデータを一括消去する機能を備えたフラッシュROMも含まれる。

【0 0 0 7】

従来、上記のEEPROMにプログラムデータを書き込む場合、2つの方法が行われていた。以下では、プログラム格納用メモリとしてフラッシュROMを備えたマイクロコンピュータを例として説明する。

【0 0 0 8】

1つの方法は、図7に示すようにROMライター110を用いてマイクロコンピュータ111に内蔵されたフラッシュROM112に平行にデータを書き込む場合である。例えば、8ビットのマイクロコンピュータ111においては、データ信号線が8本、アドレス信号線が16本、コントロール信号線が3本(チップイネーブル信号、ライトイネーブル信号、リードイネーブル信号)が必要であった。

【0 0 0 9】

また、フラッシュROM112に書き込むべきプログラムデータはパーソナル

コンピュータ内にヘキサファイル等の所定のファイル形式で存在する場合が多い。そこで、パーソナルコンピュータからフラッシュROMにプログラムデータを書き込む場合、図8に示すように、パーソナルコンピュータ100に併設されたシリアル通信ユニット115（RS232C等）を用い、シリアル信号線113を介してマイクロコンピュータ111と接続していた。

#### 【0010】

また、マイクロコンピュータ111にはSIO（Serial Input/Output）回路114が内蔵されると共に、フラッシュROM112の所定領域にはSIO回路114を動作させるためのSIO制御プログラムが予め書き込まれている。パーソナルコンピュータ100からシリアル信号線113を介してプログラムデータが転送されて来ると、SIO回路114はSIO制御プログラムに従って、フラッシュROM112に書き込み動作を行う。しかしながら、上述したシステム構成ではSIO回路114、シリアル通信ユニット115（RS232C等）という特別な外部回路と通信ソフトを必要としていた。

#### 【0011】

##### 【発明が解決しようとする課題】

上述したように、従来マイクロコンピュータに内蔵されたプログラム格納用メモリであるフラッシュROMにプログラムデータを書き込む場合、当該プログラムデータをパラレルに書き込むと信号線の本数が多くなり、シリアルに書き込む場合には特別の外部回路や通信ソフトを必要としていた。

#### 【0012】

そこで本発明は、ホスト（パーソナルコンピュータ）と周辺デバイスとを接続して双方向通信可能な環境において備えられているUSBケーブルをそのまま利用して、ホスト（パーソナルコンピュータ）からマイクロコンピュータに内蔵されたフラッシュROMへプログラムデータの書き込み等を行うと共に、プログラムデータ書き込み後はUSBケーブルを本来の目的である双方向通信に用いるようにし、特別のシリアルラインや外部回路、通信ソフト等を不要とすることを目的としている。

#### 【0013】

## 【課題を解決するための手段】

本発明のマイクロコンピュータの制御方法は、上述した課題を解決するために、ホストとマイクロコンピュータ間のデータ送受信のインターフェースを行うUSBインターフェース回路と、書き込み制御プログラムが格納された第1のプログラム領域及びプログラムデータが書き込まれる第2のプログラム領域とを有し、電氣的に書き換え及び読み出し可能な不揮発性メモリと、前記不揮発性メモリから読み出されるプログラムを実行するCPUと、を備えたマイクロコンピュータの制御方法であって、マイクロコンピュータのリセットに応じて前記書き込み制御プログラムの実行を開始する第1の段階と、USBの初期化を実行する第2の段階と、前記ホストからプログラムデータを前記不揮発性メモリの第2のプログラム領域に書き込む第3の段階と、前記第2のプログラム領域に書き込まれたプログラムを実行する第4の段階と、を有することを特徴とする。これにより、ホストからマイクロコンピュータへプログラムデータの書き込みを高速かつ円滑に行うことができる。

## 【0014】

また、前記マイクロコンピュータのリセットはパワーオンリセットであることを特徴とする。

## 【0015】

また、前記第4の段階の後に、前記ホストからプログラムデータを再書き込みするか否を判定する段階を有し、再書き込みする場合には前記第1の段階から第4の段階を再度繰り返すことにより、前記ホストからプログラムデータを前記不揮発性メモリの第2のプログラム領域に再書き込みし、当該再書き込み後のプログラムを実行することを特徴とする。これにより、ホストからのプログラムデータの再書き込みが可能となる。

## 【0016】

また、前記第3の段階の後に、前記第1のプログラム領域内の特定の領域に、照合用コードデータを書き込む段階を有すると共に、前記第1の段階の後に、前記特定の領域のデータと照合用コードデータとを比較する段階を有し、両者が一致しない場合には前記第3の段階に移行し、両者が一致した場合には前記ホスト

からプログラムデータを前記不揮発性メモリの第2のプログラム領域に再度書き込むことなく前記第2のプログラム領域に書き込まれたプログラムを実行することを特徴とする。

【0017】

これにより、プログラムデータを書き込むか、既に第2のプログラム領域に書き込まれたプログラムを実行するかの切り換えをソフト的に行うことができ、ユーザーの利便性が向上する。

【0018】

【発明の実施の形態】

次に、本発明の実施形態について図面を参照しながら説明する。図1は、本発明の実施形態に係るマイクロコンピュータの構成を示すブロック図である。

【0019】

以下で、マイクロコンピュータ10は8ビット構成として説明する。マイクロコンピュータ10とパーソナルコンピュータ100とは1対の差動信号線によって接続される。そして、USBデータ、すなわちUSB差動信号( $D^+$ ,  $D^-$ )はマイクロコンピュータ10の端子P1, P2を介して入出力される。USB差動信号( $D^+$ ,  $D^-$ )は、USB通信プロトコルに従ったシリアルデータ信号である。

【0020】

20は、端子P1, P2に接続された入出力回路であって、差動入力バッファ21、入力バッファ22, 23及び出力バッファ24, 25から構成されている。ここで、入力バッファ22, 23はUSB差動信号( $D^+$ ,  $D^-$ )の状態が(L, L)となる場合を考慮して設けられている。

【0021】

マイクロコンピュータ10に内蔵されたUSBインターフェース回路30は、パーソナルコンピュータ100との間のデータ送受信のインターフェースを行うもので、特にデータ受信時は入出力回路20からのUSB差動信号( $D^+$ ,  $D^-$ )を受けて各種のデータ処理を行う。

【0022】



USBインターフェース回路30は、上記シリアルデータ信号から必要なデータを抽出する。この時、USBインターフェース回路30は当該シリアルデータ信号が如何なる転送フォーマットであるかを判別すると共に、エラー信号処理等を行う。またUSBインターフェース回路30は、上記データ処理が施されたシリアル信号をマイクロコンピュータ100が処理可能な所定形式の平行信号（例えば8ビット構成）に変換する。

## 【0023】

さらに、USBインターフェース回路30は、平行変換された8ビット×4＝32ビットのプログラムデータを一時記憶するテンポラリレジスタ31及び制御レジスタ（不図示）を備えている。制御レジスタには、USBの初期化時にパーソナルコンピュータ100がホストとして管理すべき各種データ（デバイスに割り付けられるアドレスデータ等）がセットされる。

## 【0024】

なお、USBインターフェース回路30はマイクロコンピュータ10からパーソナルコンピュータ100へのデータ送信時には上記と全く逆のデータ処理（平行データからシリアルデータへの変換等）を行っている。

## 【0025】

RAM40はUSBインターフェース回路30のテンポラリレジスタ31のデータから逐次転送される32ビット単位のプログラムデータを一時記憶するために利用される。

## 【0026】

そして、USBインターフェース回路30とRAM40との間のデータ転送を行うために、専用の32本の信号線が設けられている。RAM40に蓄積されたプログラムデータが所定量（例えば128バイト）に達すると、128バイトのプログラムデータはマイクロコンピュータ10のバス45を経由してフラッシュROM50へ転送される。

## 【0027】

逆に、フラッシュROM50に書き込まれたプログラムデータをRAM40へ転送し、そのRAM40内に記憶されたプログラムデータをUSBインターフェ

ース回路 3 0 のテンポラリレジスタ 3 1 へ転送することも可能である。

【 0 0 2 8 】

一般に、USB 通信によればパーソナルコンピュータ 1 0 0 から大量のデータがデバイス側に送出されるため、デバイス側には特別のデータバッファを設けることが行われる。

【 0 0 2 9 】

これに対して、本発明ではマイクロコンピュータ 1 0 がデータメモリとして本来有している RAM 4 0 を USB 通信によるデータを一時記憶するために利用するという構成を採ることでデータメモリの有効活用を図っている点も特徴である。

【 0 0 3 0 】

図 2 は、RAM 4 0 及び周辺回路を示すブロック図である。USB インターフェース回路 3 0 からはアドレス信号 ADR 1、CPU 7 0 からはアドレス信号 ADR 2 が出力され、アドレス選択回路 8 0 に入力される。アドレス選択回路 8 0 はアドレス信号 ADR 1、ADR 2 のいずれかを選択してアドレス指定回路 8 1 に入力する。

【 0 0 3 1 】

そして、アドレス指定回路 8 1 の出力はアドレスデコーダ 4 1 に入力され、アドレス信号 ADR 1、ADR 2 のいずれかに応じて同一のデータ領域がアクセス可能に構成されている。

【 0 0 3 2 】

上述した構成によれば、RAM 4 0 のデータ領域 4 2 はアドレス信号 ADR 2 が選択された場合は CPU 7 0 がコントロールするデータメモリ領域として利用可能であると共に、アドレス信号 ADR 1 が選択された場合には、USB インターフェース回路 3 0 からのプログラムデータ（3 2 ビット単位）を一時記憶するためのデータメモリ領域としても利用可能である。すなわち、RAM 4 0 のデータ領域は、CPU 7 0 と USB インターフェース回路 3 0 の両方からアクセス可能である。

【 0 0 3 3 】

ただし、上記のアドレス選択は、パーソナルコンピュータ100とのデータ送受信中については、USBインターフェース回路30からのアドレス信号ADR1を選択するように構成されている。これはパーソナルコンピュータ100からのデータ転送が途中で中断できないというUSBの特性に基づくものである。具体的には、USBインターフェース回路30のテンポラリレジスタ31がフル状態になったことを検知する信号に基づいて、マイクロコンピュータ10はウェイト（待機）状態に自動的に設定される。

## 【0034】

また、図1において、50はフラッシュROMであり、USB制御プログラム（書き込み制御プログラム）が予め格納された第1のプログラム領域53とパーソナルコンピュータ100からのプログラムデータがRAM40を経由して書き込まれる第2のプログラム領域52と、に分割されている。ここで、第1のプログラム領域53は書き換えが不能なようにライトプロテクトされている。

## 【0035】

また、第2のプログラム領域52の一部には照合用コードデータ領域52Aが設けられている。この領域には、第1のプログラム領域53の書き込み制御プログラムに従って、パーソナルコンピュータ100からのプログラムデータを書き込み後に、照合用コードデータが書き込まれる。

## 【0036】

この照合用コードデータは例えば128Kバイトのデータ量であって、予め第1のプログラム領域53の一部に書き込まれている。或いは、照合用コードデータは上記プログラムデータに付随してパーソナルコンピュータ100から送信するようにしても良い。

## 【0037】

この照合用コードデータ領域52Aのデータが予め第1のプログラム領域53の一部に記憶された照合用コードデータと一致した場合には、マイクロコンピュータは既にプログラムデータが書き込み済みであると認識し、再度USBの初期化を行うことなく、当該プログラムデータを実行する。

## 【0038】

一方、この照合用コードデータ領域 5 2 A のデータが予め第 1 のプログラム領域 5 3 の一部に記憶された照合用コードデータと一致しない場合には、USB の初期化を経てプログラムデータの書き込み動作に移行する。

## 【 0 0 3 9 】

6 0 はプログラムカウンタであって、その出力はフラッシュROM 5 0 のアドレスデコーダ 5 1 に印加されている。プログラムカウンタ 6 0 の出力値は後に説明するように USB 通信の状態に応じて、CPU からの命令により所定番地にジャンプする。すなわち、パーソナルコンピュータ 1 0 0 からのプログラムデータの書き込み時には、プログラムカウンタ 6 0 は第 1 のプログラム領域 5 3（書き込み制御プログラムが格納されている）の先頭アドレスである（FF 0 0）番地にジャンプすると共に、プログラムデータの書き込み後は、第 2 のプログラム領域 5 2 の先頭アドレスである（0 0 0 0）番地にジャンプする。そして、CPU 7 0 は、フラッシュROM 5 0 から読み出されるプログラム命令に従ってマイクロコンピュータ 1 0 の動作を実行する。

## 【 0 0 4 0 】

次に、上述したマイクロコンピュータ 1 0 の制御方法について図 3 のフローチャートを参照しながら説明する。

## 【 0 0 4 1 】

まず、最初のステップ 2 0 0 では、マイクロコンピュータ 1 0 が USB ケーブルに接続される。このとき、USB ケーブルの電源ラインによってマイクロコンピュータ 1 0 に電源が投入されることにより、マイクロコンピュータ 1 0 は内蔵されたパワーオンリセット回路によりリセットされる。

## 【 0 0 4 2 】

次に、ステップ 2 0 1 において、書き込み制御プログラムが開始する。このとき、プログラムカウンタ 6 0 の値は、第 1 のプログラム領域 5 3 の先頭アドレスである（FF 0 0）番地へジャンプする。従って、その後マイクロコンピュータ 1 0 は当該書き込み制御プログラムに従って以下の処理を実行する。

## 【 0 0 4 3 】

ステップ 2 0 1 において USB ケーブルにマイクロコンピュータ 1 0 が接続さ

れると、マイクロコンピュータ 1 0 側に設けられたプルアップ抵抗を介して、U S B 差動信号 ( $D^+$ ,  $D^-$ ) が (L, L) から例えば (H, L) が変化する。パーソナルコンピュータ 1 0 0 はこの U S B 差動信号 ( $D^+$ ,  $D^-$ ) の変化により、マイクロコンピュータ 1 0 0 が U S B ネットワークに接続されたことを検知し、所定時間後に U S B バスリセット信号を発行する。U S B バスリセット信号の受信後、次のステップに進む。

## 【 0 0 4 4 】

ステップ 2 0 3 では、エニミュレーション (Enumeration) による U S B の初期化を行う。これは、一般にマイクロコンピュータ 1 0 とパーソナルコンピュータ 1 0 0 との間で U S B データの送受信を行うことが可能な環境設定を行うための一連のソフトウェア処理である。

## 【 0 0 4 5 】

エニミュレーションにより行われる主な処理は、パーソナルコンピュータ 1 0 0 の初期化と、パーソナルコンピュータ 1 0 0 が支配するデバイスにアドレスを割り付ける処理である。後者において、U S B インターフェース回路 3 0 内の制御レジスタ (アドレスレジスタ) 内に、パーソナルコンピュータ 1 0 0 が割り当てた特定のアドレスが記憶される。これにより、マイクロコンピュータ 1 0 は、パーソナルコンピュータ 1 0 0 が送信して来た U S B パケット内のアドレスと上記アドレスレジスタ内のアドレスとを照合し、それらが一致した場合にのみ送信されきた U S B データの処理を行う。

## 【 0 0 4 6 】

こうして、U S B データの送受信を行うことが可能な環境設定が終了すると、ステップ 2 1 0 では、パーソナルコンピュータ 1 0 0 からのプログラムデータのロードを実行する。

## 【 0 0 4 7 】

このステップ 2 1 0 の詳細は図 4 のフローチャートに示す。ステップ 2 1 1 では、パーソナルコンピュータ 1 0 0 からフラッシュ R O M に書き込むべきプログラムデータが U S B 差動信号データ ( $D^+$ ,  $D^-$ ) の形で入力されてくる。

## 【 0 0 4 8 】

ステップ 2 1 2 では、この入力された USB 差動信号データ ( $D^+$ ,  $D^-$ ) を USB インターフェース回路 3 0 によってデータ処理する。このデータ処理内容は上述した通りであるが、シリアルデータ (8 ビット  $\times$  4) を所定のパラレルデータ (3 2 ビット) に変換するのがその主な処理である。

## 【 0 0 4 9 】

ステップ 2 1 3 では、USB インターフェース回路 3 0 から RAM 4 0 へパラレル変換されたプログラムデータが書き込まれる。そして、RAM 4 0 へ書き込まれたプログラムデータ量が所定量 (例えば 1 2 8 バイト) に達すると、この所定量を単位として RAM 4 0 からバス 4 5 を介してフラッシュ ROM 5 0 の第 1 の領域へ書き込みが開始される (ステップ 2 1 4)。これはフラッシュ ROM 5 0 が複数ブロックに分割されており、1 2 8 バイトをブロックとして構成されていることによる。したがって、RAM 4 0 のデータの蓄積量はフラッシュ ROM 5 0 のブロック構成に応じて適宜に選択可能である。

## 【 0 0 5 0 】

ここで、実際には USB インターフェース回路 3 0 から RAM 4 0 へパラレル変換されたプログラムデータの書き込み動作と、RAM 4 0 からフラッシュ ROM 5 0 へ書き込み動作は並行して行われるために、高速書き込みが実現される。

## 【 0 0 5 1 】

ステップ 2 1 4 において、フラッシュ ROM への書き込みが開始されるがこれには所定の時間を要する。そこで、ステップ 2 1 5 ではマイクロコンピュータ 1 0 はソフト的に NACK 状態にセットされる。これは USB パケットのハンドシェイク・パケットの一種であって、ホストであるパーソナルコンピュータ 1 0 0 からのデータを受け付けることができないことを知らせるためにパーソナルコンピュータ 1 0 0 へ返される。

## 【 0 0 5 2 】

そして、ステップ 2 1 6 では書き込み終了か否かを判定する。その判定結果が NO であれば、NACK 状態を維持する。その判定結果が YES であれば、ACK 状態にセットされ (ステップ 2 1 7)、ACK はマイクロコンピュータ 1 0 側でデータを受け付け可能であることを知らせるためにパーソナルコンピュータ 1

0 0 へ返される。

【 0 0 5 3 】

そして、次のステップ 2 1 8 ではフラッシュROM 5 0 へのプログラムデータの書き込みが全て終了したかを判定する。その判定結果が N O であれば、ステップ 2 1 1 へ戻り、残余のプログラムデータの書き込みを続行する。ここで、プログラムデータの書き込みはブロック（ページ）単位（例えば 1 2 8 バイト）で行われるため、全部のページが書き込まれるまでこの処理は繰り返される。

【 0 0 5 4 】

判定結果が Y E S の場合には、上記の過程で書き込まれたプログラムを実行する（ステップ 2 2 0）。すなわち、プログラムカウンタ 6 0 の値は第 2 のプログラム領域 5 2 の先頭アドレスである（0 0 0 0）番地にジャンプする。そして、マイクロコンピュータ 1 0 はパーソナルコンピュータ 1 0 から供給されたプログラムデータを読み出し、C P U 7 0 は解読されたプログラム命令に基づいてマイクロコンピュータの動作を実行開始する。

【 0 0 5 5 】

なお、上記のようにパーソナルコンピュータ 1 0 0 からマイクロコンピュータ 1 0 のフラッシュROM 5 0 に対してプログラムデータを書き込む場合のプログラムステップについて説明したが、フラッシュROM 5 0 に書き込まれたプログラムデータを読み出して、パーソナルコンピュータ 1 0 0 へ送り返し、ベリファイを行うことも可能である。

【 0 0 5 6 】

その場合には、プログラムステップの順序は上述したものと逆の順序となる。すなわち、フラッシュROM 5 0 から読み出されたプログラムデータはRAM 4 0 に一時記憶された後、U S B インターフェース回路 3 0 へ逐次転送される。そして、U S B インターフェース回路 3 0 では書き込みの際とは逆のデータ処理を施し、パラレルデータを所定のシリアルデータに変換後、U S B ケーブルを介してパーソナルコンピュータ 1 0 0 へ送出する。

【 0 0 5 7 】

図 3 のフローチャートに戻ると、パーソナルコンピュータ 1 0 0 から書き込ま

れたプログラムの実行の過程（ステップ 2 2 0）において、再度、プログラムデータを書き込むかどうかを判定する（ステップ 2 2 1）。これは、USBケーブルの断線等の何らかの通信エラーに対応するためのステップである。そこで、YESと判定された場合には、ステップ 2 0 3に戻り、プログラムデータの再ロードが行われる。

#### 【 0 0 5 8 】

このように、上述したマイクロコンピュータの制御方法によれば、フラッシュROM 5 0の第 1のプログラム領域 5 3に予め記憶された書き込み制御プログラムに従って、パーソナルコンピュータ 1 0 0からプログラムデータを第 1のプログラム領域 5 3に書き込むと共に、当該プログラムを実行することが可能となる。

#### 【 0 0 5 9 】

さて、実使用状態において、マイクロコンピュータ 1 0からUSBケーブルをはずし電源オフしてから、再度USBケーブルを接続し電源オンにする場合を考えると、既に実行すべきプログラムデータがロードされているにもかかわらず、上記図 3のプログラムフローに従うと、書き込み制御プログラムが起動してしまい、不要なプログラムステップが実行されてしまうという不都合が生じる。これを回避するには外部のジャンパスイッチによって 2つのプログラムを切り分ける方法が考えられるが、ユーザーにとって非常に不便である。

#### 【 0 0 6 0 】

そこで、図 5のフローチャートに示すように、書き込み制御プログラムの実行の過程で、プログラムデータの書き込み判定を行うステップ 2 0 2を設けた。すなわち、第 2のプログラム領域 5 2の一部には照合用コードデータ領域 5 2 Aが設けられている。

#### 【 0 0 6 1 】

この照合用コードデータ領域 5 2 Aには、プログラムデータの書き込み後（ステップ 2 1 8）に、照合用コードデータが書き込まれる。前述したように、照合用コードデータは例えば 1 2 8 K バイトのデータ量であって、予め第 1のプログラム領域 5 3の一部に書き込まれていても良いし、或いはプログラムデータに付



随してパーソナルコンピュータ 1 0 0 から送信するようにしても良い。

【 0 0 6 2 】

ステップ 2 0 2 では、照合用コードデータ領域 5 2 A のデータを読み出し、予め第 1 のプログラム領域 5 3 に書き込まれているか、マイクロコンピュータ内の所定のレジスタに記憶された照合用コードデータと比較する。それらが一致した場合には、マイクロコンピュータは既にプログラムデータが書き込み済みであると認識し、再度 U S B の初期化を行うことなく、当該プログラムデータを実行する。

【 0 0 6 3 】

一方、照合用コードデータ領域 5 2 A のデータが照合用コードデータと一致しない場合には、U S B の初期化を経てプログラムデータの書き込み動作に移行する。これにより、2 つのプログラムの切り分けをソフトウェアにより行うことが可能となるのでユーザーの利便性が向上する。

【 0 0 6 4 】

また、バージョンアップされたプログラムデータを再書き込みしたい場合には、上記の照合用コードデータ領域 5 2 A の照合コードデータを書き換えるか、その 1 ビットを削除する等により、照合判定結果は不一致となるので、再ロードすることが可能となる。

【 0 0 6 5 】

【発明の効果】

本発明のマイクロコンピュータの制御方法によれば、ホスト（例えばパーソナルコンピュータ）と周辺デバイスとの接続のための U S B ケーブルを利用して、ホストからマイクロコンピュータへのプログラム書き込みを高速に行うことが可能となる。また、プログラムの転送に U S B を利用しているので、特別のシリアルラインや外部回路、通信ソフト等を不要とすることができる。

【 0 0 6 6 】

また、本発明によれば、不揮発性メモリの一部領域に照合用コードデータを書き込み、プログラムデータが書き込み済みかどうか書き込み照合判定を行っているので、書き込み制御プログラムと実行すべきプログラムの切り分けをソフトウ

エアにより行うことが可能となるのでユーザーの利便性が向上するという利点を有している。

【図面の簡単な説明】

【図 1】

本発明の実施形態に係るマイクロコンピュータを示すブロック図である。

【図 2】

本発明の実施形態に係るマイクロコンピュータの R A M 及び周辺回路を示すブロック図である。

【図 3】

本発明の実施形態に係るマイクロコンピュータ 1 0 の制御方法を示すフローチャートである。

【図 4】

本発明の実施形態に係るマイクロコンピュータ 1 0 の制御方法を示すフローチャートである。

【図 5】

本発明の実施形態に係るマイクロコンピュータ 1 0 の制御方法を示すフローチャートである。

【図 6】

U S B を利用したパーソナルコンピュータと周辺デバイスとの接続構成例を示す図である。

【図 7】

従来のフラッシュ R O M へのプログラムデータ書き込み方法を示す図である。

【図 8】

従来のフラッシュ R O M へのプログラムデータ書き込み方法を示す図である。

【符号の説明】

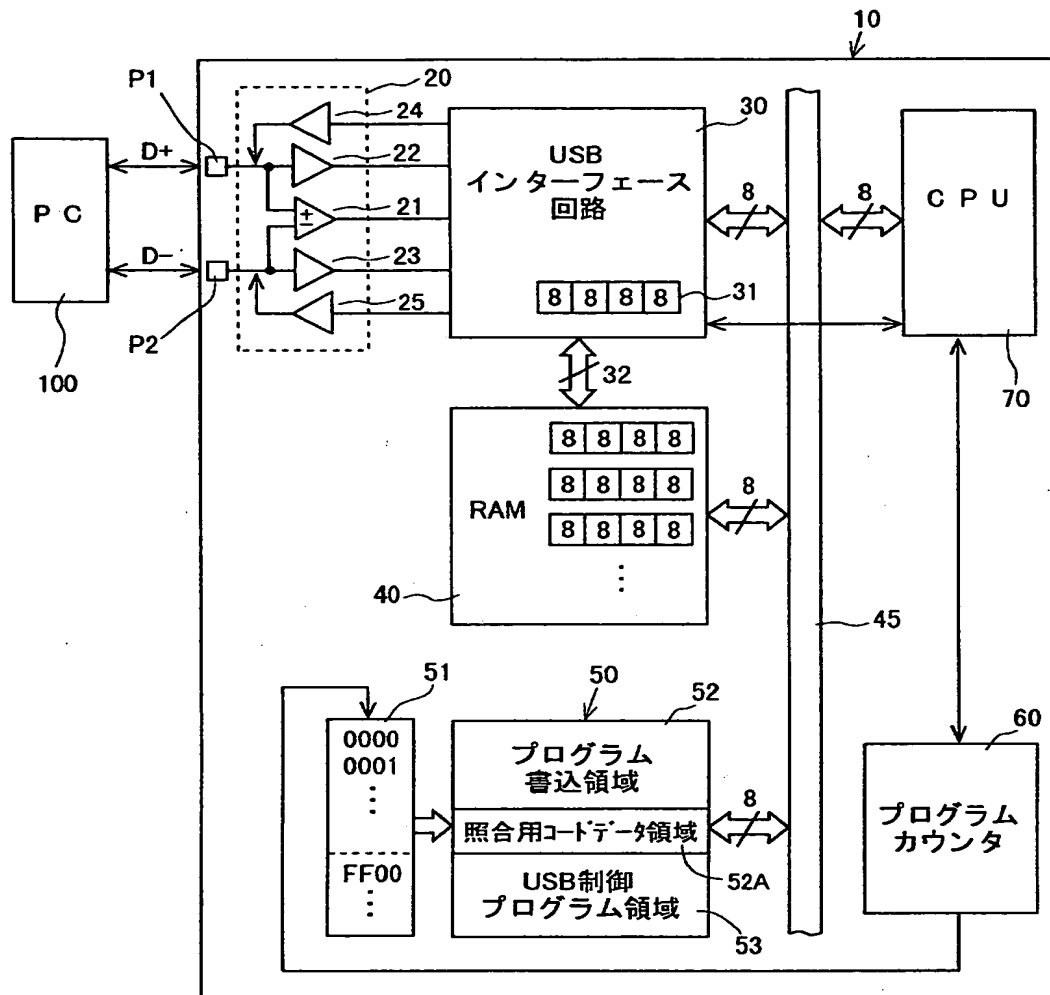
- 1 0     マイクロコンピュータ
- 2 0     入出力回路
- 3 0     U S B インターフェース回路
- 4 0     R A M

- 4 5    バス
- 5 0    フラッシュROM
- 5 1    アドレスデコーダ
- 5 2    第2のプログラム領域
- 5 3    第1のプログラム領域
- 6 0    プログラムカウンタ
- 7 0    CPU

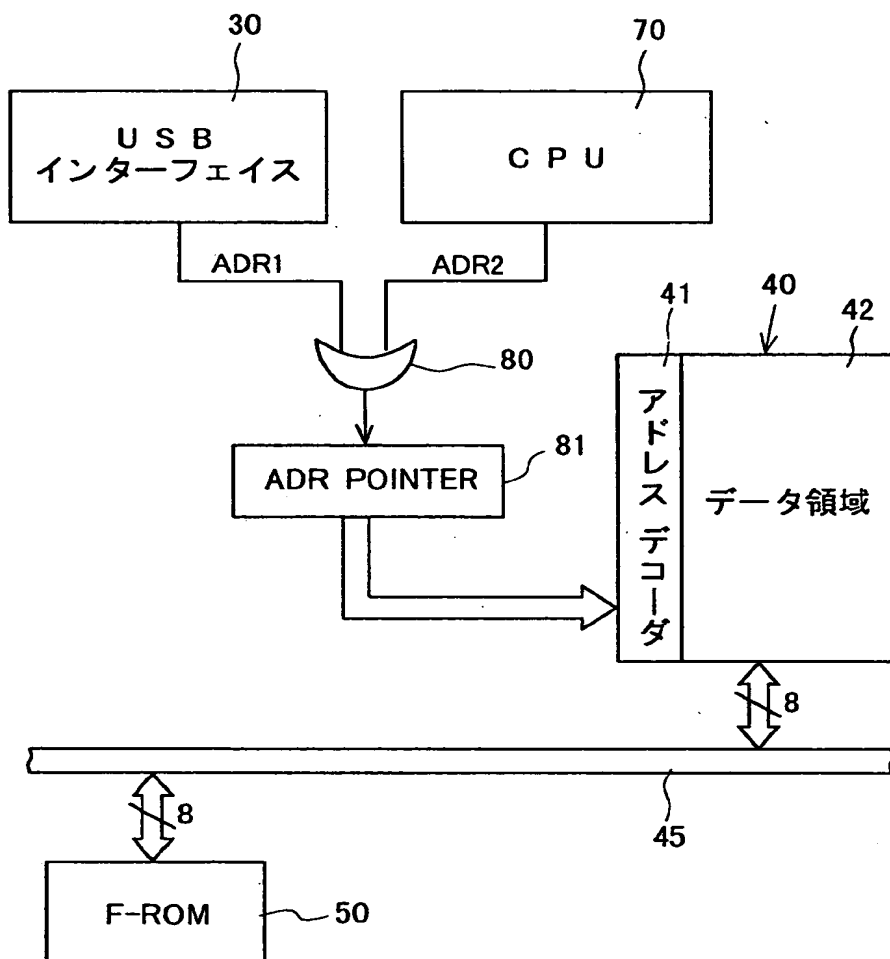
【書類名】

図面

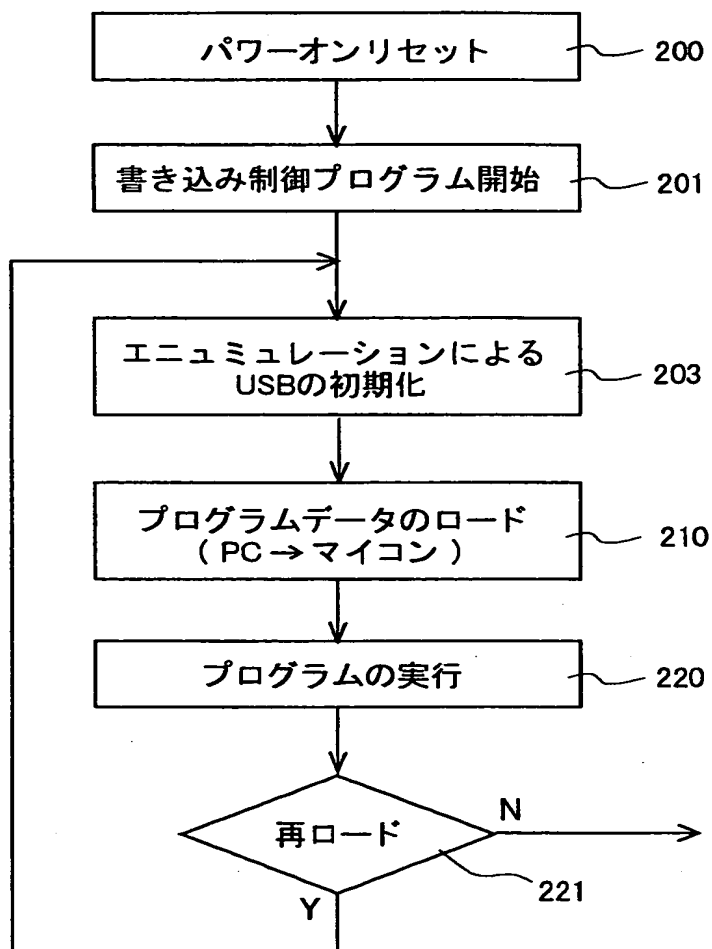
【図 1】



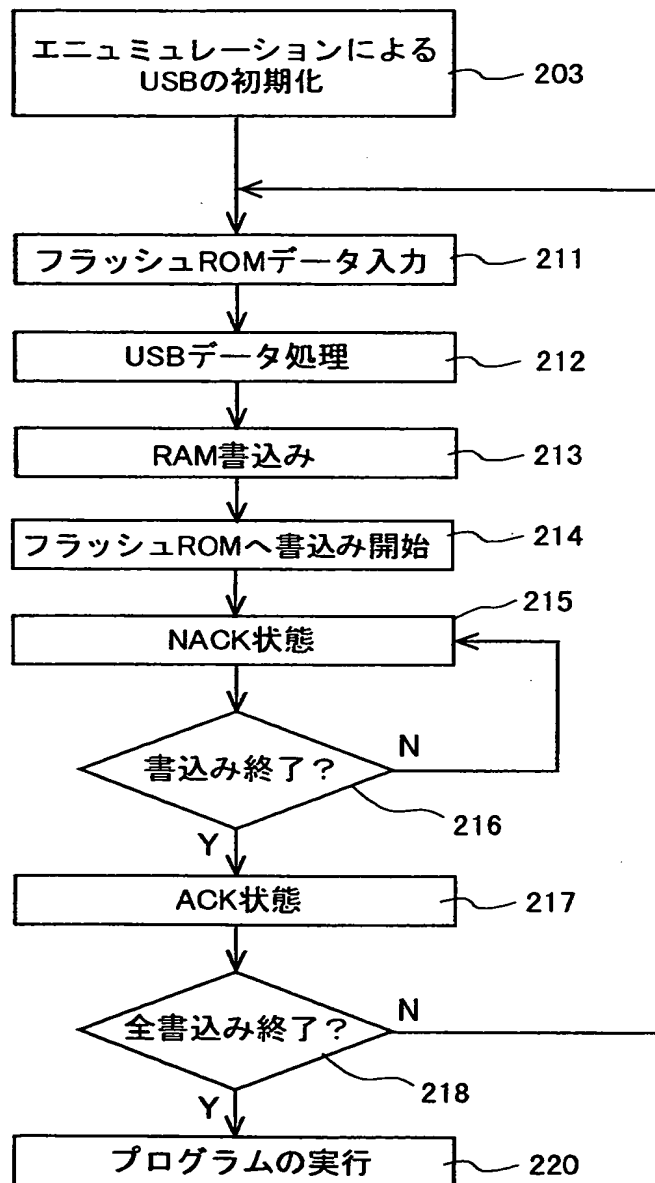
【図 2】



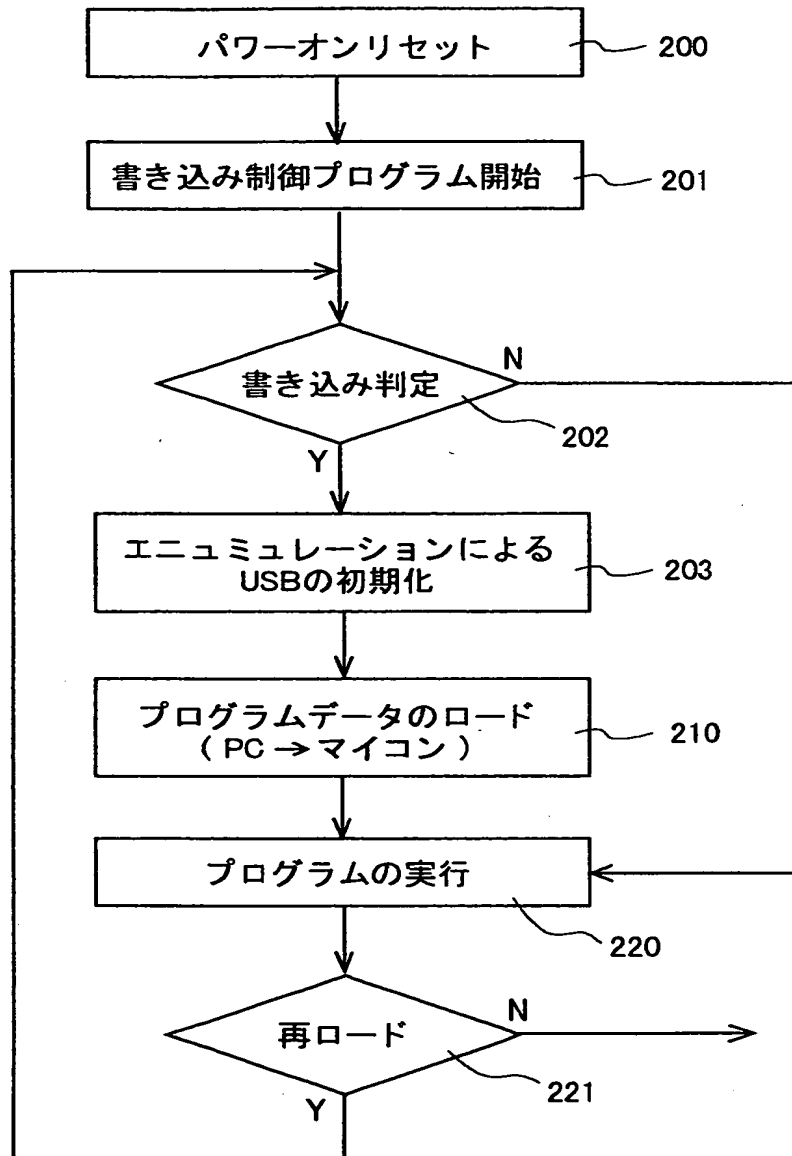
【図 3】



【図 4】

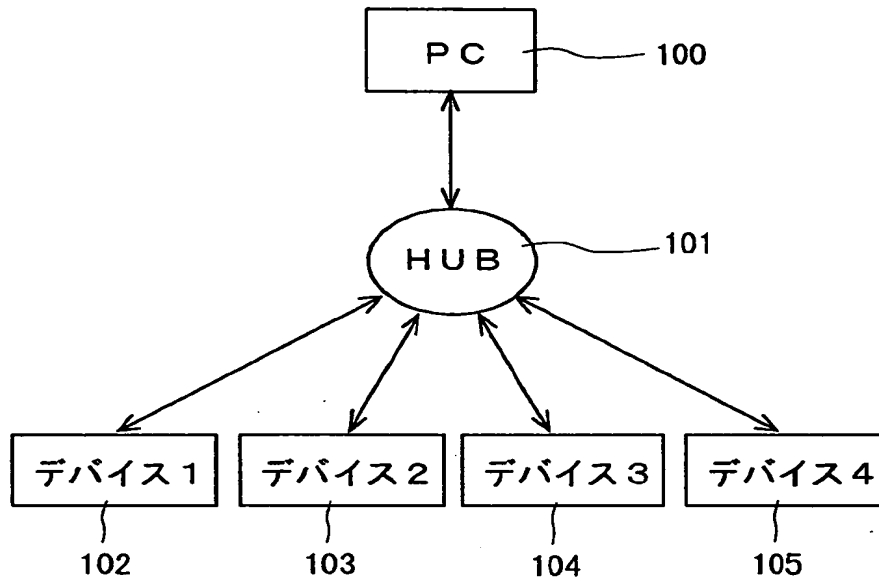


【図 5】

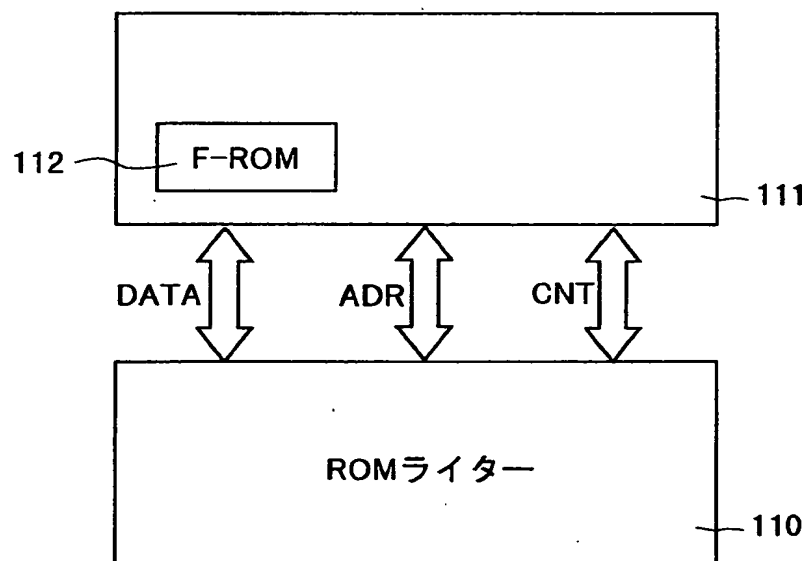




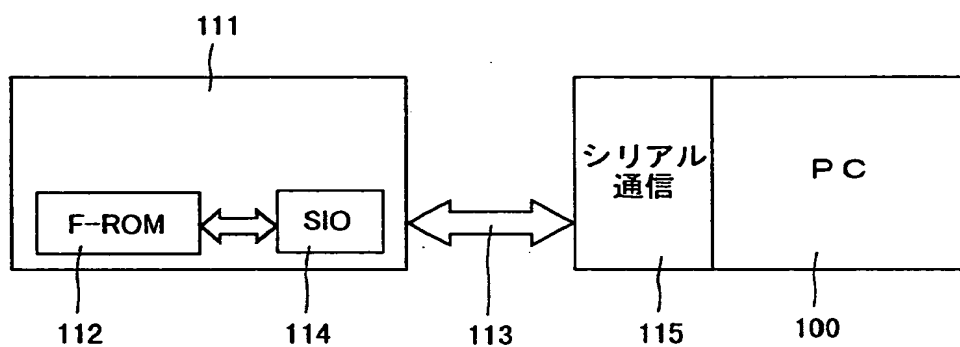
【図 6】



【図 7】



【図 8】



【書類名】 要約書

【要約】

【課題】 ホストと周辺デバイスとの接続のためのUSBケーブルを利用して、ホストからマイクロコンピュータへのプログラム書き込みを高速に行う。

【解決手段】 マイクロコンピュータのリセットに応じて前記書き込み制御プログラムの実行を開始する第1のステップ201と、前記書き込み制御プログラムによってUSBの初期化を実行する第2のステップ203と、USBの初期化後、前記ホストからプログラムデータを前記不揮発性メモリの第1のプログラム領域に書き込む第3のステップ210と、前記第1のプログラム領域に書き込まれたプログラムを実行する第4のステップ220とを有する。

【選択図】 図3

出 願 人 履 歴 情 報

識別番号 [000001889]

1. 変更年月日	1993年10月20日
[変更理由]	住所変更
住 所	大阪府守口市京阪本通2丁目5番5号
氏 名	三洋電機株式会社